

# **Rapport Détaillé sur le Déploiement de l'Application HumansBestFriend avec Docker**

**Membre du groupe :**

Airouche Kafia

Ouamrane Lydia

## **Sommaire :**

### **1. Vue d'Ensemble**

- **Présentation de HumansBestFriend**
- **Technologies utilisées (Python, Node.js, .NET, Redis, Postgres)**
- **Rôle de Docker et Docker Compose**

### **2. Mise en Route**

- **Fork et Clone du Répertoire**
  - **Fork du projet**
  - **Clone sur machine virtuelle Linux**

### **3. Étapes de Déploiement**

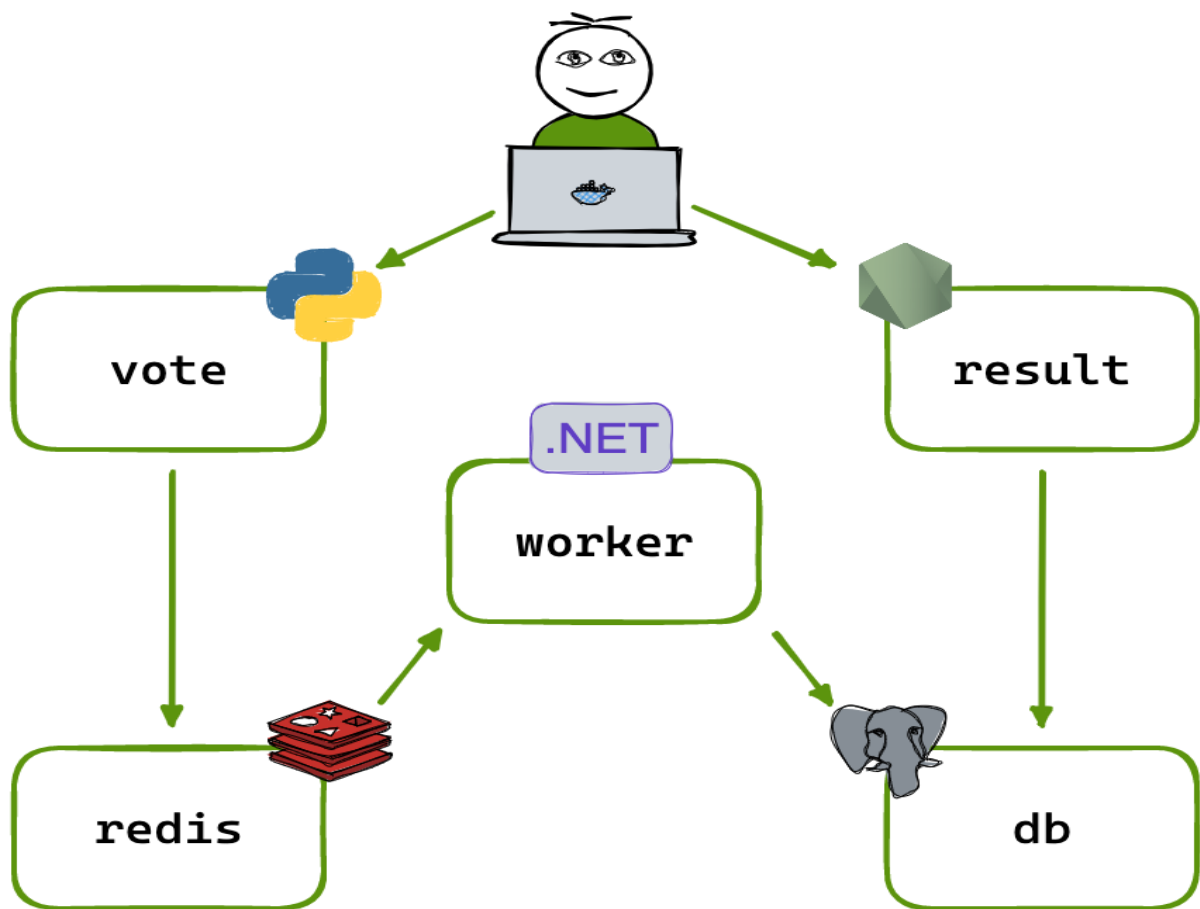
- **Création d'un Docker Compose Build**
  - **Élaboration du fichier docker-compose.build.yml**
  - **Exécution et vérification des images Docker**
- **Création d'un Registre**
  - **Mise en place du conteneur de registre Docker**
- **Étiquetage des Images**
  - **Renommage de l'image Postgres**
  - **Étiquetage pour le registre privé**
- **Pousser les Images**
  - **Envoi des images au registre**
  - **Vérification des images dans le registre**
- **Création du Fichier Docker Compose**
  - **Configuration du fichier compose**
- **Création du Réseau**
  - **Établissement du réseau humansbestfriend-network**
- **Exécution du Docker Compose**
  - **Déploiement de l'application**

### **4. Résultats**

### **5. Conclusion**

## Vue d'Ensemble :

HumansBestFriend est une application distribuée conçue pour fonctionner à travers plusieurs conteneurs Docker. Ce document fournit un guide complet sur la configuration et le déploiement de l'application en utilisant Docker et Docker Compose. L'application intègre diverses technologies, y compris Python, Node.js, .NET, avec Redis pour la messagerie et Postgres pour le stockage de données.



## Mise en Route :

### 1. Fork et Clone du Répertoire :

- Fork du projet à partir du dépôt Git du prof.

- Clone du dépôt dans la machine virtuelle Linux pour le développement et le déploiement.

```

kaf@admin:~$ git clone https://github.com/LydiaOuam/ynov-resources.git
Cloning into 'ynov-resources'...
remote: Enumerating objects: 408, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 408 (delta 4), reused 26 (delta 2), pack-reused 376
Receiving objects: 100% (408/408), 11.55 MiB | 7.56 MiB/s, done.
Resolving deltas: 100% (59/59), done.

```

## Étapes de Déploiement :

### 1. Création d'un Docker Compose Build

- **Création du Fichier Compose:**

- Créez un fichier nommé docker-compose.build.yml.
- Le fichier peut être trouvé [ici](https://github.com/LydiaOuam/ynov-resources/blob/main/2023/m2/dataeng/humans-best-friend/docker-compose.build.yml).  
(<https://github.com/LydiaOuam/ynov-resources/blob/main/2023/m2/dataeng/humans-best-friend/docker-compose.build.yml>)
- Exécutez le fichier avec la commande :

`docker-compose -f docker-compose.build.yml up`

```

kaf@admin:~/ynov-resources/2023/m2/dataeng$ git pull
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 1), reused 6 (delta 1), pack-reused 0
Unpacking objects: 100% (6/6), 572 bytes | 190.00 KiB/s, done.
From https://github.com/LydiaOuam/ynov-resources
 ffe7584..efd7cdb  main -> origin/main
Updating ffe7584..efd7cdb
Fast-forward
 2023/m2/dataeng/docker-compose.build.yml | 8 ++++----
 1 file changed, 4 insertions(+), 4 deletions(-)
kaf@admin:~/ynov-resources/2023/m2/dataeng$ docker-compose -f docker-compose.build.yml up
Creating network "dataeng-default" with the default driver
Building result
[+] Building 8.1s (4/11)
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 525B
=> [internal] load .dockerignore
=> => transferring context: 54B
=> [internal] load metadata for docker.io/library/node:18-slim
=> [1/7] FROM docker.io/library/node:18-slim@sha256:fe687021c06383a2bc5eafa6db29b627ed28a55f6bdfbcea108f0c624b783c37
=> => resolve docker.io/library/node:18-slim@sha256:fe687021c06383a2bc5eafa6db29b627ed28a55f6bdfbcea108f0c624b783c37
=> => sha256:8e04602828dddb8c394c781f265c048f2a7cf9c8bf112635ba26cec2d069236f17b 1.37kB / 1.37kB
=> => sha256:d3cce7487840f2783532a374ca9e79c4d55fca3d5ca3ca47c145596fcc457f9c 7.62kB / 7.62kB
=> => sha256:af107e978371b6cd6339127a05502c5eacd1e6b0e9eb7b2f4aa7b6fc97e2dd81 22.02MB / 29.13MB
=> => sha256:ebd7ac832a7e4bd0775ad6f4bb6c6db1f5b02de7a8e07b40627142ef61a9b1b9d 3.36kB / 3.36kB
=> => sha256:fe687021c06383a2bc5eafa6db29b627ed28a55f6bdfbcea108f0c624b783c37 1.21kB / 1.21kB
=> => sha256:9d5fc5b38df6ebdd70895cf78cd4e3de9aef9dc55ed5c79c2945066661a0e8e4 18.87MB / 38.57MB
=> => sha256:e4cb19787d8cf84786456582936c59a7c23c841a29861d728df3f6c8923dd739 2.67MB / 2.67MB
=> => sha256:6f59eaf6ede7c1fe4ad170ff96c351c02b84f6a0a7d4f38dcbal3d4642298b 452B / 452B
=> [internal] load build context
=> => transferring context: 278.07kB

```

- **Vérifiez la création des images :**

- docker images

```
kaf@admin:~/ynov-resources/2023/m2/dataeng/humans-best-friend$ docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
seed-data           latest          a76accabee810  5 hours ago    129MB
seed                latest          a76accabee810  5 hours ago    129MB
localhost:5000/seed-data latest          a76accabee810  5 hours ago    129MB
localhost:5000/seed latest          a76accabee810  5 hours ago    129MB
worker              latest          973664ed08fd   5 hours ago    194MB
localhost:5000/worker latest          973664ed08fd   5 hours ago    194MB
vote                latest          f30bfef5ba50   5 hours ago    154MB
localhost:5000/vote latest          f30bfef5ba50   5 hours ago    154MB
result              latest          914cb3b0a0c2   5 hours ago    224MB
localhost:5000/result latest          914cb3b0a0c2   5 hours ago    224MB
postgres            15-alpine       c94362bdb5ee   7 days ago     240MB
localhost:5000/db   latest          c94362bdb5ee   7 days ago     240MB
redis               latest          e40e2763392d   2 weeks ago    138MB
localhost:5000/redis latest          e40e2763392d   2 weeks ago    138MB
registry            2               909c3ff012b7   2 weeks ago    25.4MB
kaf@admin:~/ynov-resources/2023/m2/dataeng/humans-best-friend$
```

## 2. Création d'un Registre

- **Conteneur de Registre:**

- Exécutez la commande suivante pour créer un conteneur de registre :
- `docker run -d -p 5000:5000 --restart always --name registry registry:2`

```
kaf@admin:~/ynov-resources/2023/m2/dataeng$ docker run -d -p 5000:5000 --restart always --name registry registry:2
Unable to find image 'registry:2' locally
2: Pulling from library/registry
c926b61bad3b: Pull complete
5501dced60f8: Pull complete
e875fe5e6b9c: Pull complete
21f4bf2f86f9: Pull complete
98513cca25bb: Pull complete
Digest: sha256:0a182cb82c93939407967d6d71d6caf11dcef0e5689c6afe2d60518e3b34ab86
Status: Downloaded newer image for registry:2
fc26e3b2b41dcdec824a4ed4a5f987d60114b4df5927f4949e2a7500105a3554
kaf@admin:~/ynov-resources/2023/m2/dataeng$
```

## 3. Étiquetage des Images

- **Renommer l'Image Postgres:**

- Renommez l'image Postgres avec la commande :

- `docker tag postgres db`

### Étiquetage pour le Registre Privé:

- Étiquetez toutes les images avec `localhost:5000` pour les pousser vers le registre privé :

```
kaf@admin:~/ynov-resources/2023/m2/dataeng$ docker tag db localhost:5000/db
kaf@admin:~/ynov-resources/2023/m2/dataeng$ docker tag seed localhost:5000/seed
kaf@admin:~/ynov-resources/2023/m2/dataeng$ docker tag redis localhost:5000/redis
kaf@admin:~/ynov-resources/2023/m2/dataeng$ docker tag worker localhost:5000/worker
kaf@admin:~/ynov-resources/2023/m2/dataeng$ docker tag result localhost:5000/result
kaf@admin:~/ynov-resources/2023/m2/dataeng$ docker tag vote localhost:5000/vote
```

## 4. Pousser les Images

- **Pousser vers le Registre:**
  - Pousser les images étiquetées vers le registre :

```
kaf@admin:~/ynov-resources/2023/m2/dataeng$ docker push localhost:5000/db
Using default tag: latest
The push refers to repository [localhost:5000/db]
902cdae7f91e: Pushed
157ec5ba867b: Pushed
77b88378eb76: Pushed
57ae9058d30b: Pushed
fc3f887ab385: Pushed
9fb828a65405: Pushing [=====>] 91.16MB/307.8MB
c71db60c1bd0: Pushed
31875ba284ec: Pushed
5fd5147dd181: Pushed
18c1be6a9b1e: Pushed
167497dbb0e6: Pushed
87e76904e32f: Pushed
d6943465a5bb: Pushed
7292cf786aa8: Pushing [=====>] 22.26MB/74.77MB
```

Vérifiez les images dans le registre :

- `curl localhost:5000/v2/_catalog`

```
latest: digest: sha256:db9760c720b4b38c71b7b0bc27e334d3201da8021197b001b073c97002acc410-31
kaf@admin:~/ynov-resources/2023/m2/dataeng$ curl localhost:5000/v2/_catalog
{"repositories":["db","redis","result","seed","vote","worker"]}
kaf@admin:~/ynov-resources/2023/m2/dataeng$
```

## 5. Créer le Fichier Docker Compose

- **Configuration du Fichier Compose:**
  - Créez le fichier Docker Compose conformément aux exigences spécifiées.
  - Le fichier Docker Compose peut être consulté ici.  
(<https://github.com/LydiaOuam/ynov-resources/blob/main/2023/m2/dataeng/humans-best-friend/compose.yml>)

## 6. Création du Réseau

- **Création du Réseau:**
  - Créez un réseau nommé `humansbestfriend-network` :

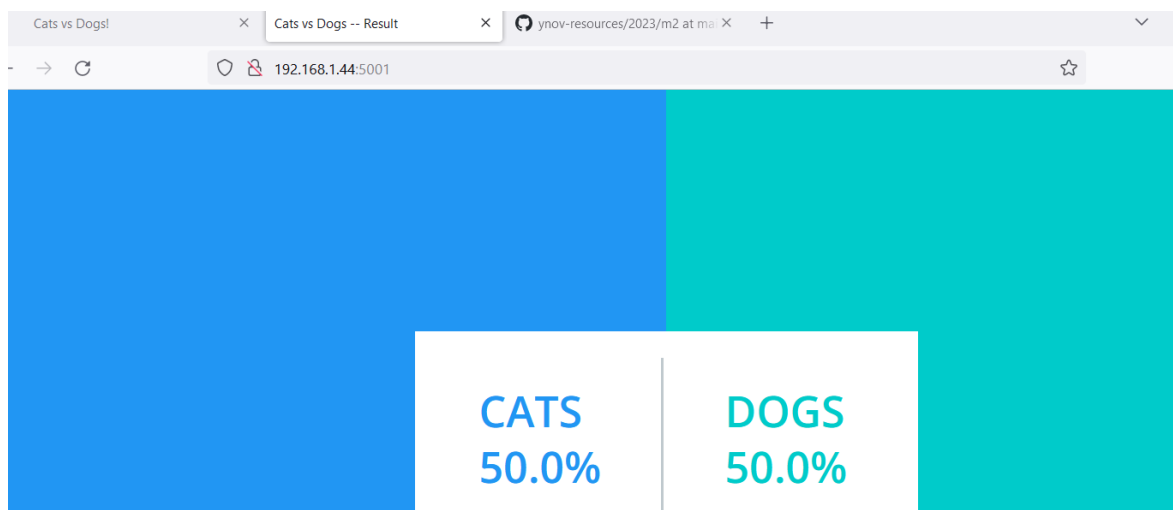
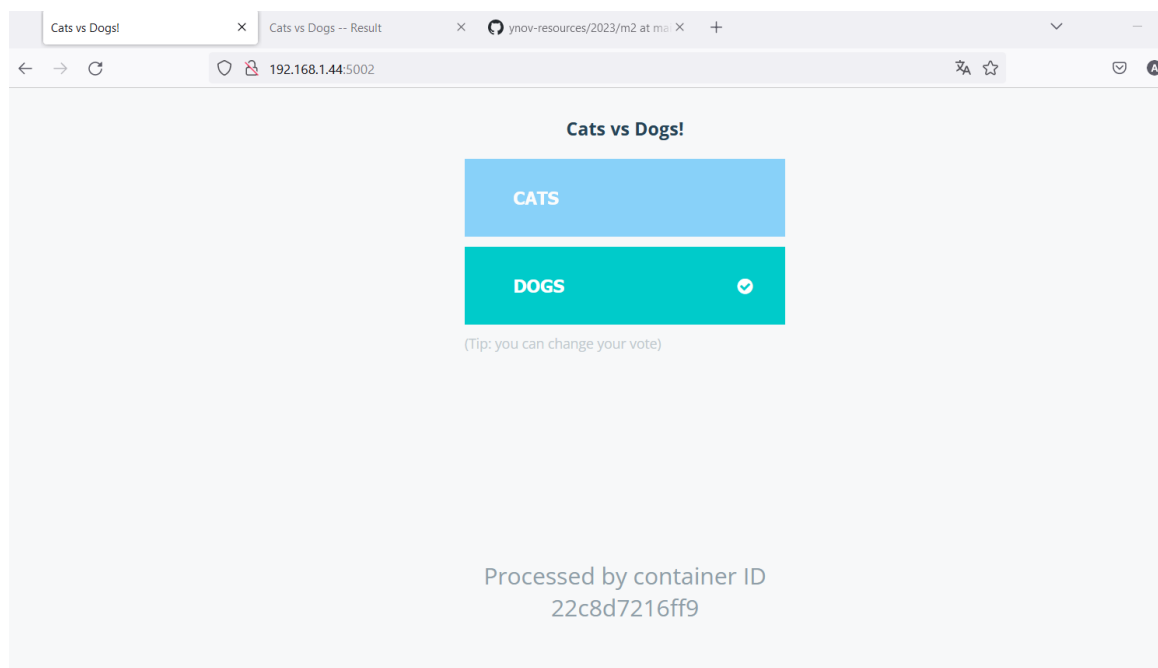
```
kaf@admin:~/ynov-resources/2023/m2/dataeng$ docker network create humansbestfriend-network
0bafb516f7f410351a62f3c073dd22d0de68d2e1233d0e9e6a4f7e96f0a562ff
kaf@admin:~/ynov-resources/2023/m2/dataeng$
```

## 7. Exécuter le Docker Compose

- **Déploiement de l'Application:**
  - Exécutez le fichier Docker Compose : `docker-compose up`

```
kaf@admin:~/ynov-resources/2023/m2/dataeng/humans-best-friend$ docker compose up
[+] Running 6/6
✓ Network humans-best-friend_humansbestfriend-network Created 0.2s
✓ Container humans-best-friend-db-1 Created 0.2s
✓ Container humans-best-friend-vote-1 Created 0.3s
✓ Container humans-best-friend-redis-1 Created 0.2s
✓ Container humans-best-friend-result-1 Created 0.1s
✓ Container humans-best-friend-worker-1 Created 0.1s
Attaching to humans-best-friend-db-1, humans-best-friend-redis-1, humans-best-friend-result-1, humans-best-friend-vote-1, humans-best-friend-worker-1
humans-best-friend-db-1 | The files belonging to this database system will be owned by user "postgres".
humans-best-friend-db-1 | This user must also own the server process.
humans-best-friend-db-1 |
humans-best-friend-db-1 | The database cluster will be initialized with locale "en_US.utf8".
humans-best-friend-db-1 | The default database encoding has accordingly been set to "UTF8".
humans-best-friend-db-1 | The default text search configuration will be set to "english".
humans-best-friend-db-1 |
humans-best-friend-db-1 | Data page checksums are disabled.
humans-best-friend-db-1 |
humans-best-friend-db-1 | fixing permissions on existing directory /var/lib/postgresql/data ... ok
humans-best-friend-db-1 | creating subdirectories ... ok
humans-best-friend-db-1 | selecting dynamic shared memory implementation ... posix
humans-best-friend-db-1 | selecting default max_connections ... 100
humans-best-friend-db-1 | selecting default shared_buffers ... 128MB
humans-best-friend-db-1 | selecting default time zone ... UTC
humans-best-friend-db-1 | creating configuration files ... ok
```

## Résultats :



## Tester la connexion entre les deux containers :

```
kaf@admin:~/ynov-resources/2023/m2/dataeng/humans-best-friend$ docker exec -it 28fb8832ddfe bash
root@28fb8832ddfe:/usr/local/app# apt-get update
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [52.1 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 Packages [8787 kB]
Get:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [11.3 kB]
Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages [128 kB]
Fetched 9177 kB in 3s (3526 kB/s)
Reading package lists... Done
root@28fb8832ddfe:/usr/local/app# apt-get install iputils-ping
Reading package lists... Done
Building dependency tree... Done

root@28fb8832ddfe:/usr/local/app# ping -c 5 22c8d7216ff9
PING 22c8d7216ff9 (172.31.0.2) 56(84) bytes of data:
64 bytes from humans-best-friend-vote-1.humans-best-friend_humansbestfriend-network (172.31.0.2): icmp_seq=1 ttl=64 time=0.625 ms
64 bytes from humans-best-friend-vote-1.humans-best-friend_humansbestfriend-network (172.31.0.2): icmp_seq=2 ttl=64 time=0.059 ms
64 bytes from humans-best-friend-vote-1.humans-best-friend_humansbestfriend-network (172.31.0.2): icmp_seq=3 ttl=64 time=0.063 ms
64 bytes from humans-best-friend-vote-1.humans-best-friend_humansbestfriend-network (172.31.0.2): icmp_seq=4 ttl=64 time=0.080 ms
64 bytes from humans-best-friend-vote-1.humans-best-friend_humansbestfriend-network (172.31.0.2): icmp_seq=5 ttl=64 time=0.102 ms

--- 22c8d7216ff9 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4063ms
rtt min/avg/max/mdev = 0.059/0.185/0.625/0.220 ms
```

## Conclusion :

Ce document décrit les étapes nécessaires au déploiement de l'application HumansBestFriend en utilisant Docker et Docker Compose. En suivant ces étapes, l'application sera configurée et opérationnelle dans un environnement distribué à travers plusieurs conteneurs, exploitant la puissance de Docker pour un déploiement et une mise à l'échelle efficaces.